Design Optimization of Time- and Cost-Constrained Fault-Tolerant Distributed Embedded Systems

Izosimov, Viacheslav; Pop, Paul; Eles, Petru; Peng, Zebo

Published in:
Design Optimization of Time- and Cost-Constrained Fault-Tolerant Distributed Embedded Systems

Link to article, DOI:
10.1109/DATE.2005.116

Publication date:
2005

Document Version
Publisher's PDF, also known as Version of record

Design Optimization of Time- and Cost-Constrained Fault-Tolerant Distributed Embedded Systems

Viaceslav Izosimov, Paul Pop, Petru Eles, Zebo Peng
Embedded Systems Lab (ESLAB)
Linköping University, Sweden

Motivation

- Hard real-time applications
  - Timing constraints
  - Cost constraints

- **Faults**
  - Predictable
  - **Transient**
  - Intermittent

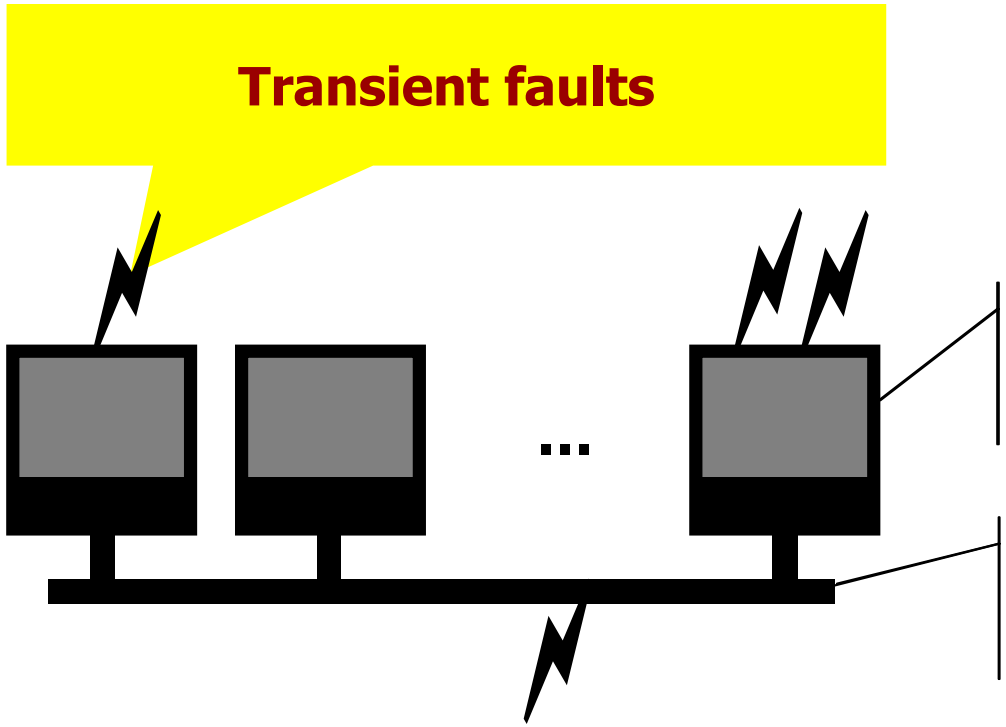
- Hardware solutions **vs.** **Software solutions**
  - MARS, TTA, X-by-Wire
    - Permanent faults
    - Costly for transient faults
  - Re-execution/rollback recovery
  - Checkpointing/rollback recovery
  - Replication, primary-backup…

- Online preemptive **vs.** **Off-line non-preemptive**
  - Flexible
  - Predictable

Outline

- Motivation
- ➔ System architecture and fault-model
  - Fault-tolerance techniques
- Problem formulation
  - Motivational examples
- Tabu-search optimization strategy
- Experimental results
- Contributions and Message

Fault-Tolerant Time-Triggered Systems

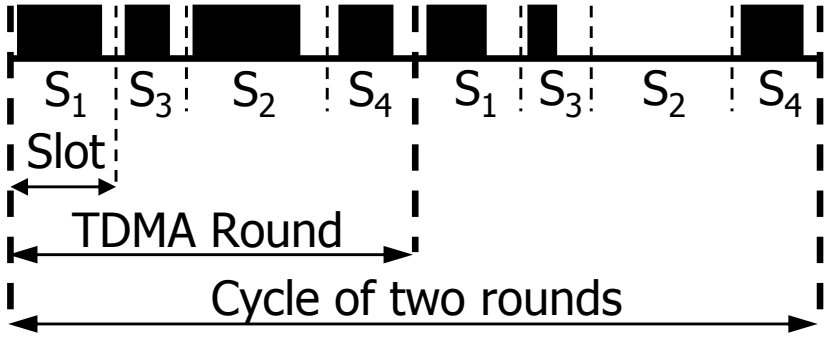


Processes:
Re-execution and replication

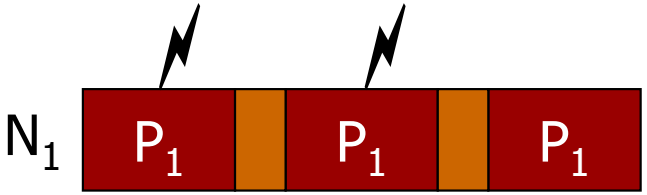
Messages:
Fault-tolerant protocol

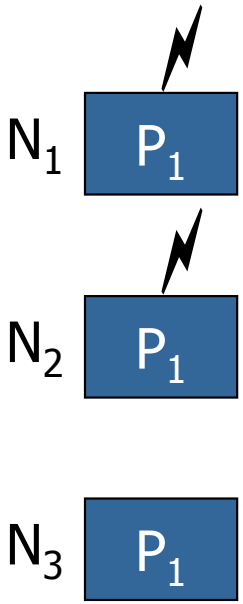
Time Triggered Protocol (**TTP**)

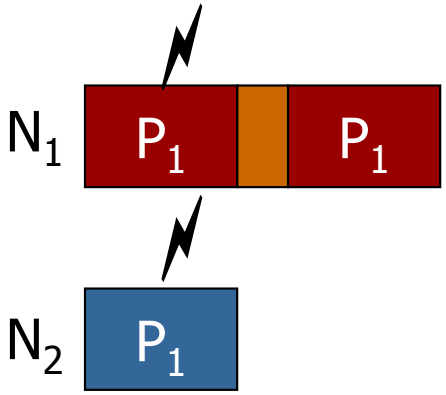
- Bus access scheme:
  time-division multiple-access (TDMA)
- Schedule table located in each TTP
  controller: message descriptor list (MEDL)



Fault-Tolerant Techniques







Re-execution

Replication

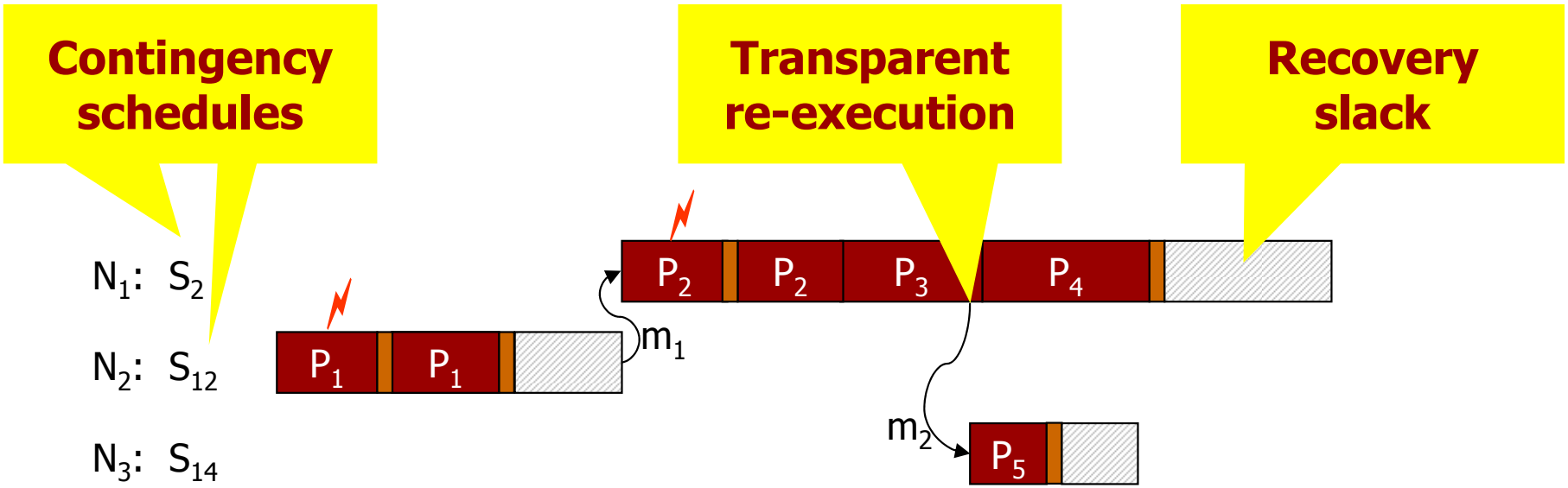
Re-executed replicas

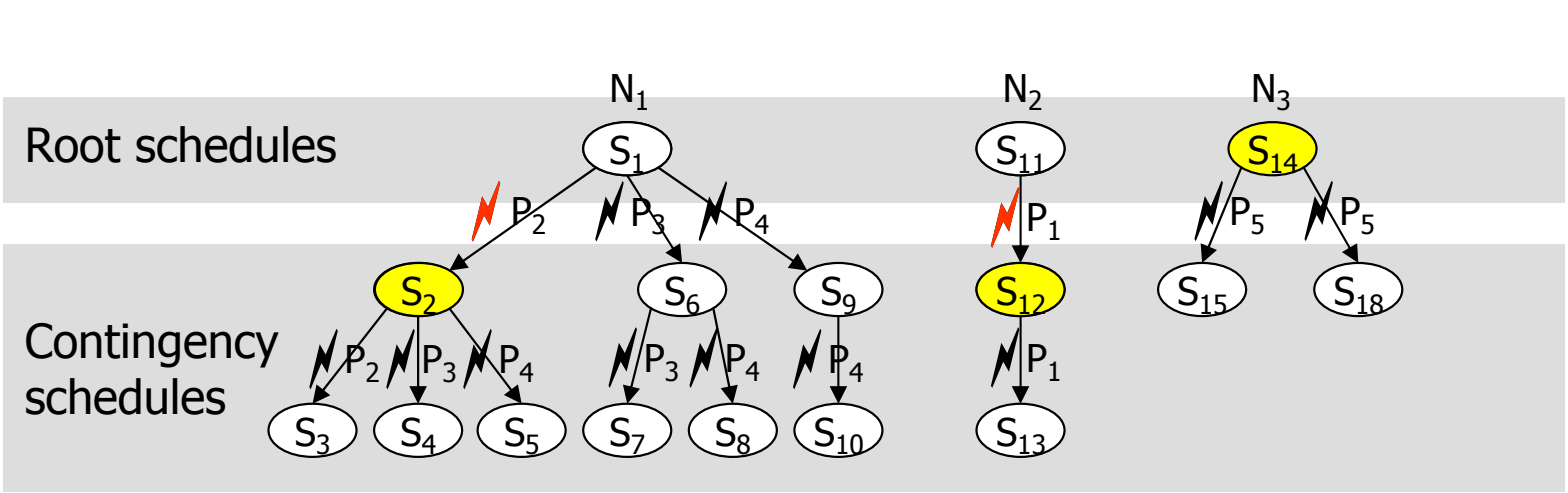
Problem Formulation

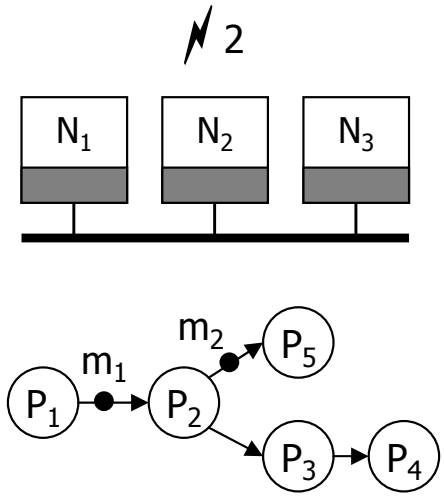
- Given
  - **Fault model**
    - Number of transient faults in the system period
  - System architecture
  - Application
    - WCETs, message sizes, periods, deadlines

- Determine
  - **Schedulable** and **fault-tolerant** design implementation
    - Fault-tolerance policy assignment
    - Mapping of processes and messages
    - Schedule tables for processes and messages

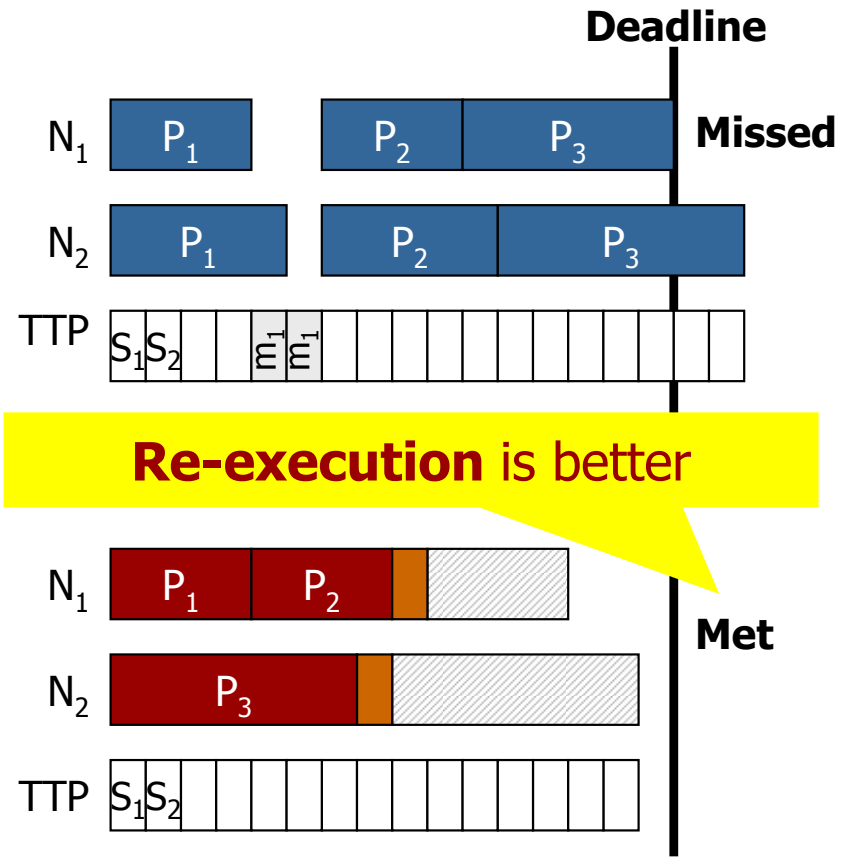
Static Scheduling [Kandasamy et al. 03]

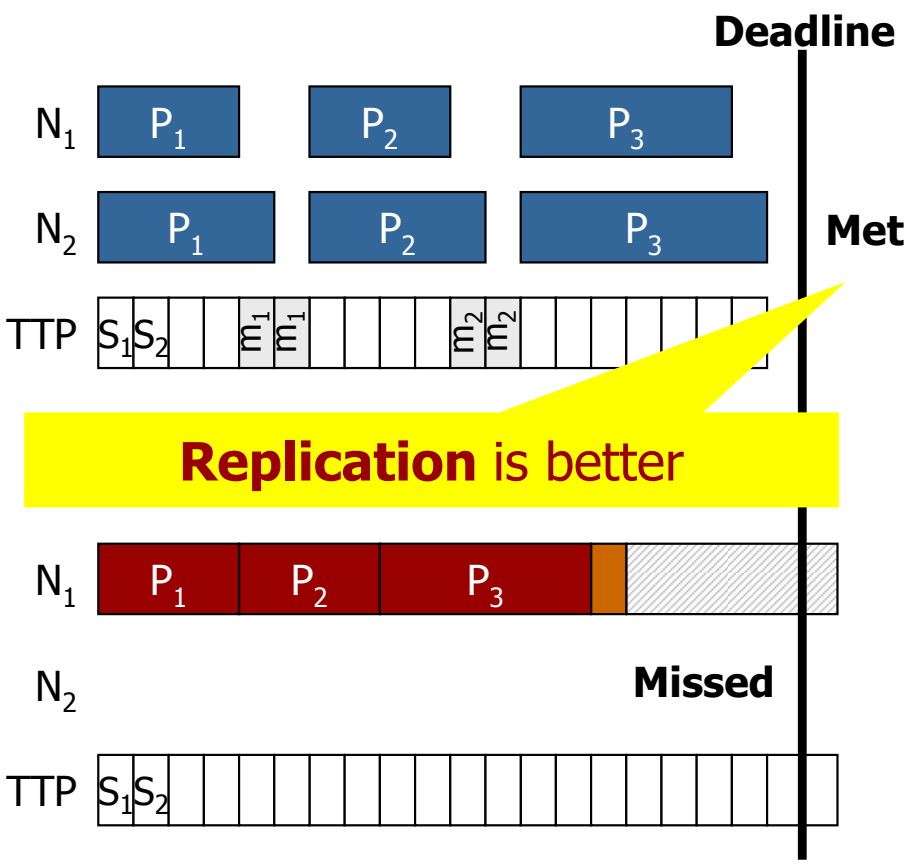


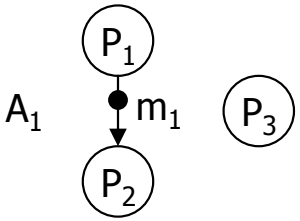


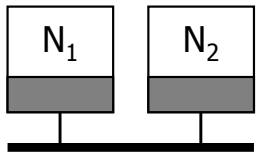


Re-execution vs. Replication



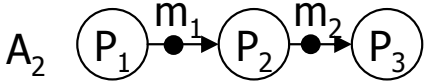




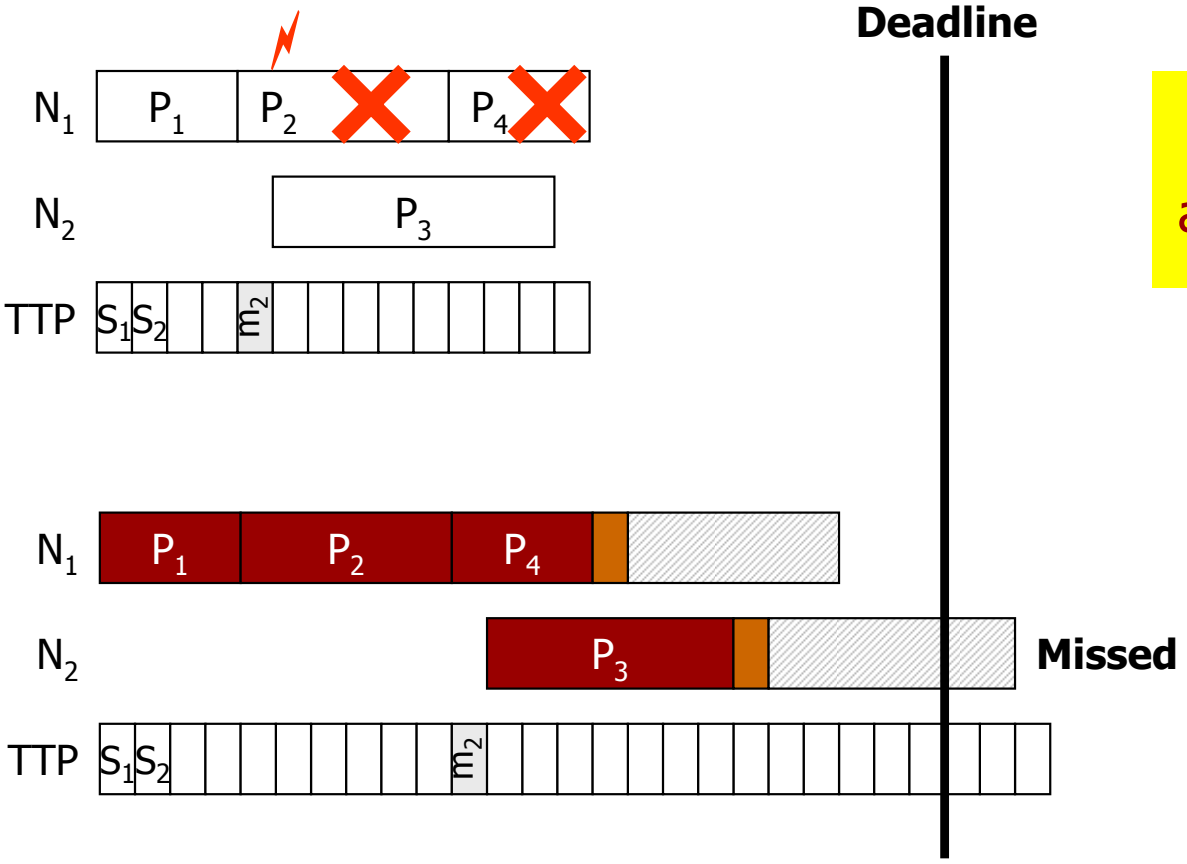


	N_1	N_2
P_1	40	50
P_2	40	50
P_3	60	70

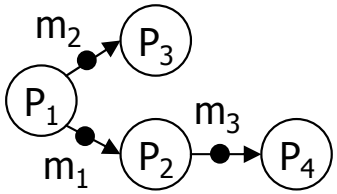
1



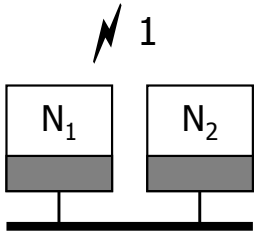
Fault-Tolerant Policy Assignment



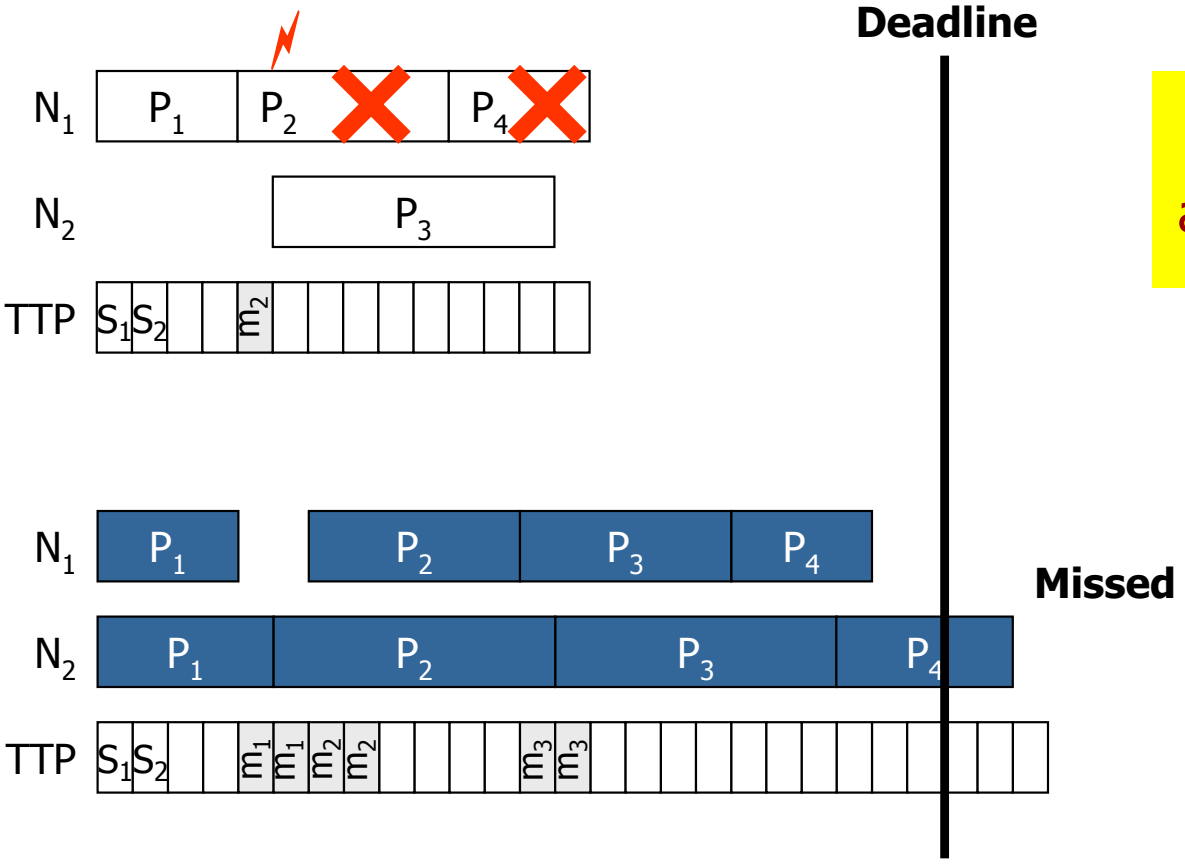
No fault-tolerance: application **crashes**



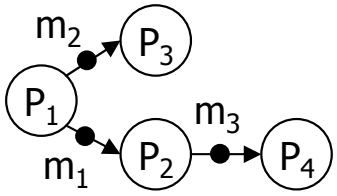
	N_1	N_2
P_1	40	50
P_2	60	80
P_3	60	80
P_4	40	50



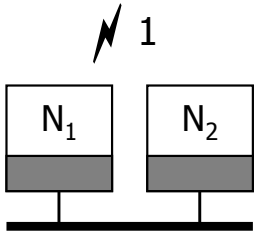
Fault-Tolerant Policy Assignment



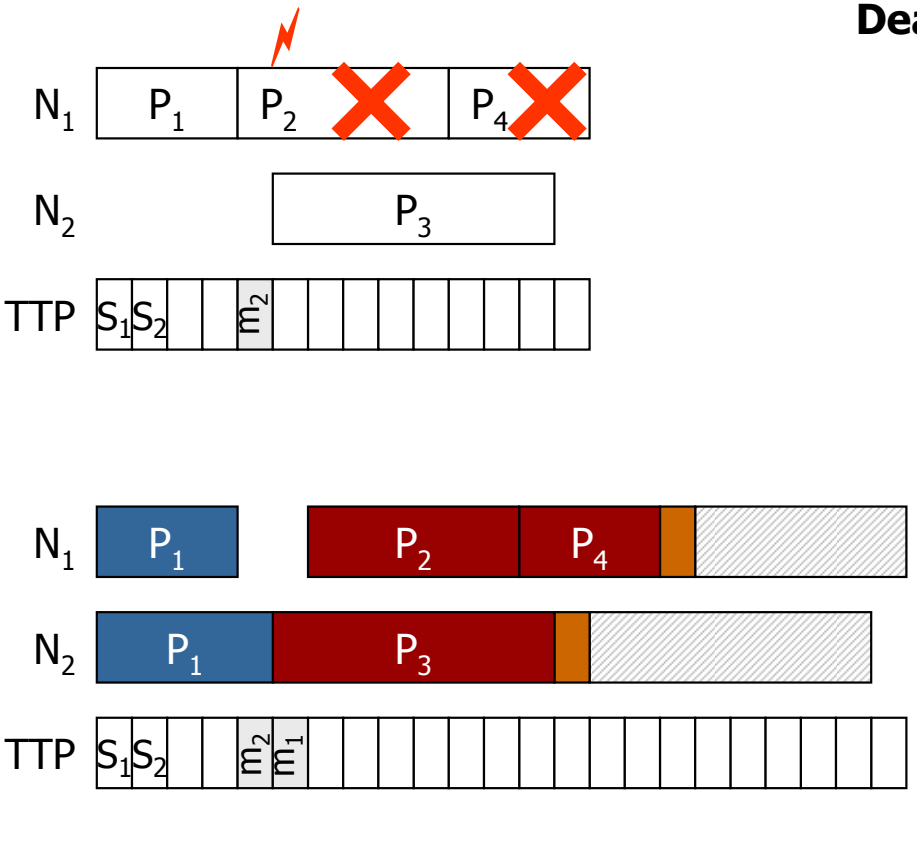
No fault-tolerance: application **crashes**



	N_1	N_2
P_1	40	50
P_2	60	80
P_3	60	80
P_4	40	50

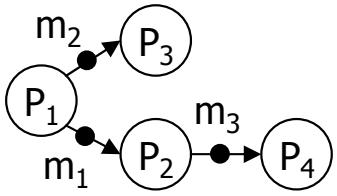


Fault-Tolerant Policy Assignment

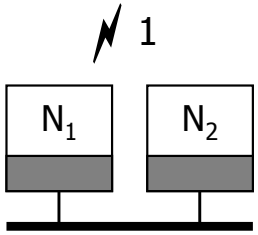


No fault-tolerance: application **crashes**

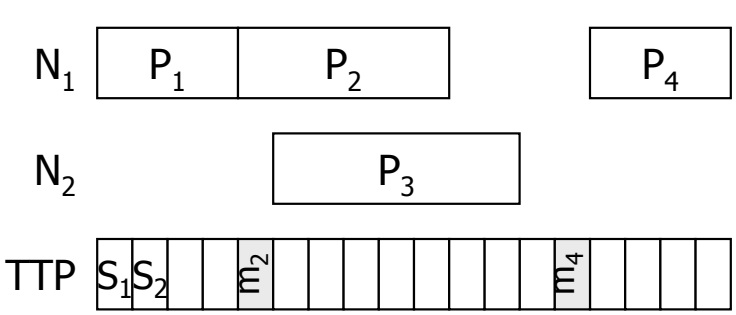
Optimization of fault-tolerance policy assignment



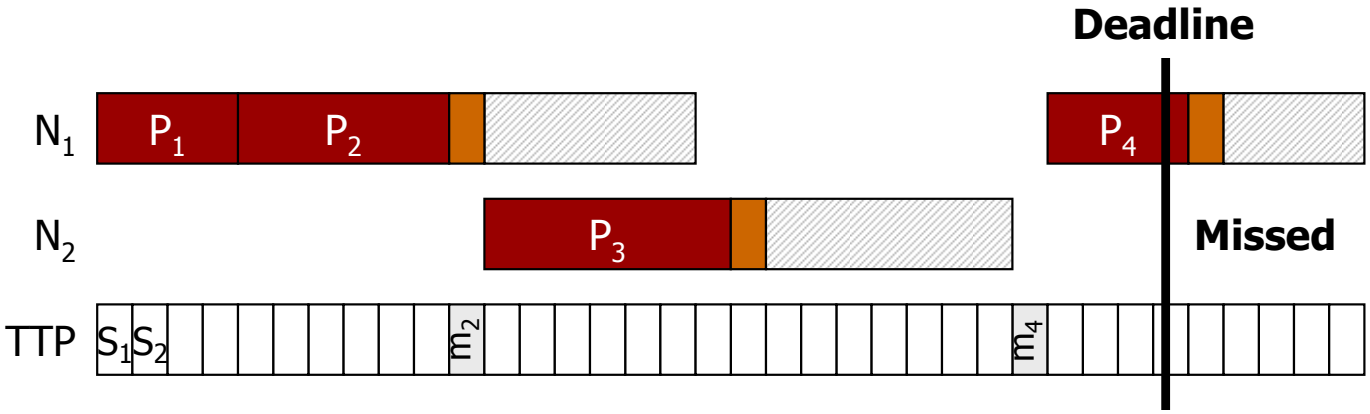
	N_1	N_2
P_1	40	50
P_2	60	80
P_3	60	80
P_4	40	50

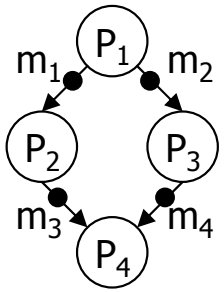


Mapping and Fault-Tolerance

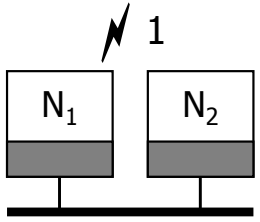


Best mapping without considering fault-tolerance

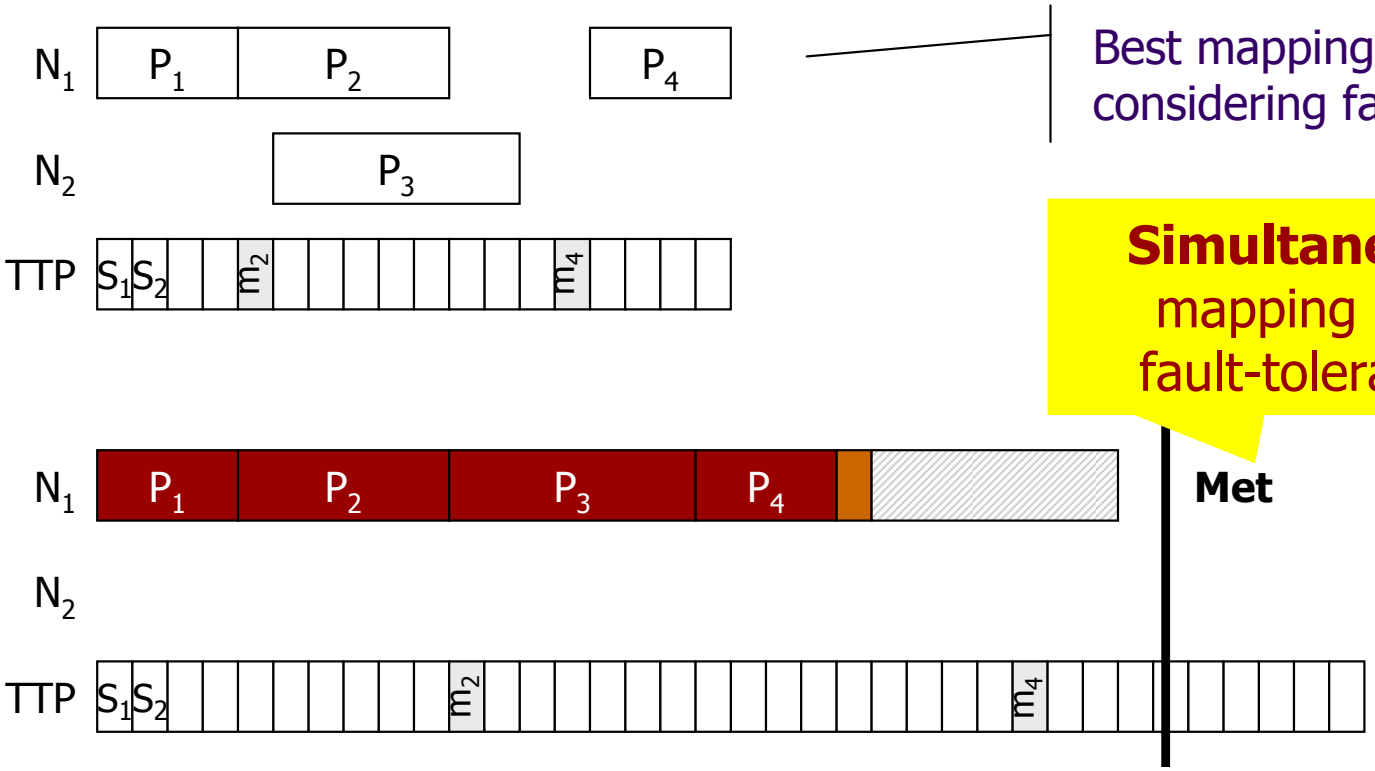


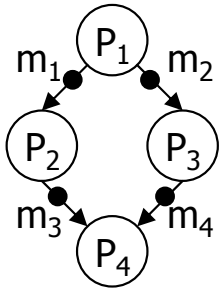


	N_1	N_2
P_1	40	X
P_2	60	70
P_3	60	70
P_4	40	X

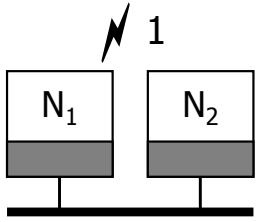


Mapping and Fault-Tolerance





	N_1	N_2
P_1	40	X
P_2	60	70
P_3	60	70
P_4	40	X



Optimization Strategy

- Design optimization:
  - Fault-tolerance policy assignment
  - Mapping of processes and messages

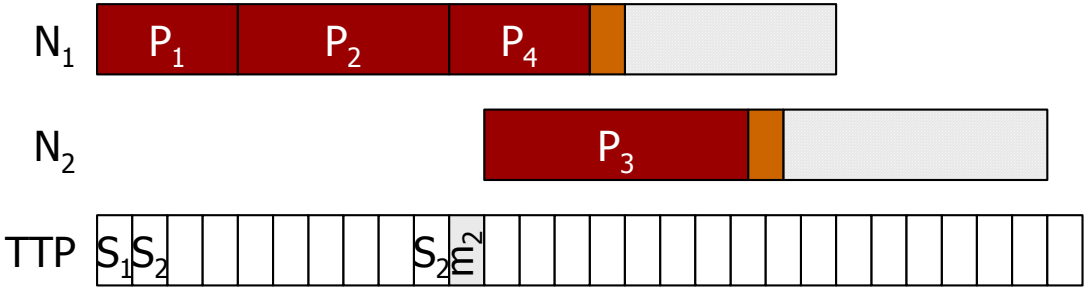
  } **Tabu-search**

  - Root schedules

  } **List scheduling**

- Three tabu-search optimization algorithms:
  1. Mapping and Fault-Tolerance Policy assignment (**MRX**)
     - Re-execution, replication or both
  2. Mapping and only Re-Execution (**MX**)
  3. Mapping and only Replication (**MR**)

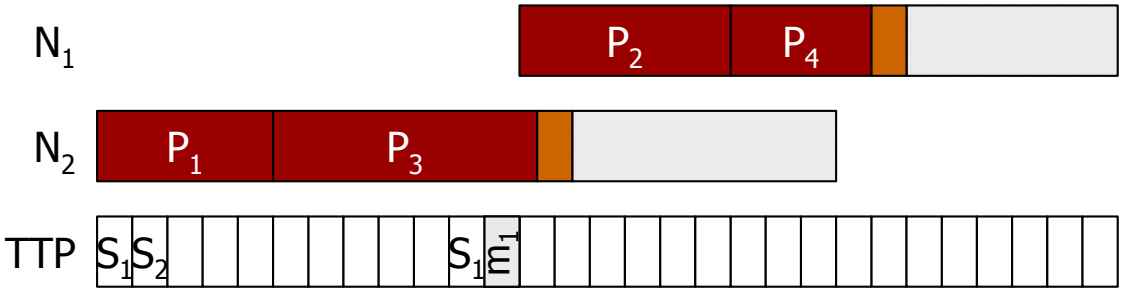
MRX Tabu-Search Example



	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

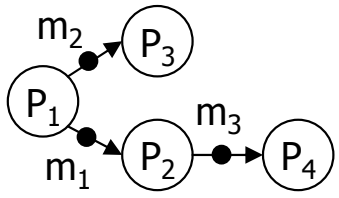
Current solution

Design transformations

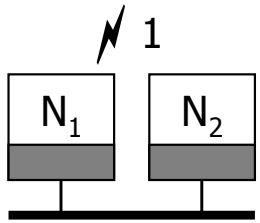


	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

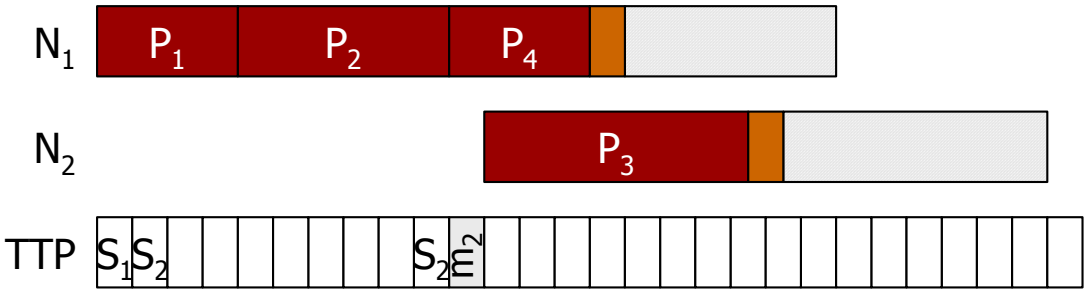
Tabu move & **worse** than best-so-far



	N_1	N_2
P_1	40	50
P_2	60	75
P_3	60	75
P_4	40	50



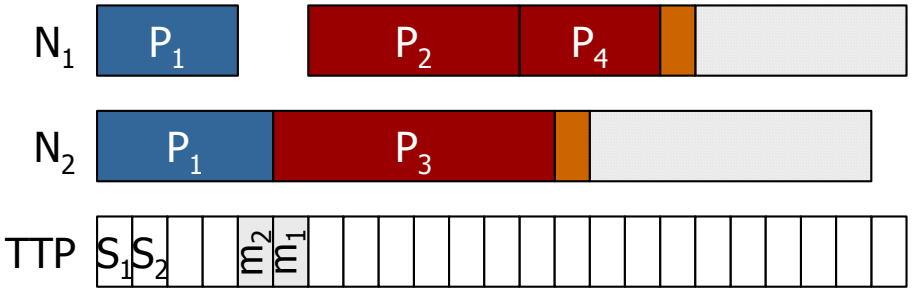
MRX Tabu-Search Example



	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

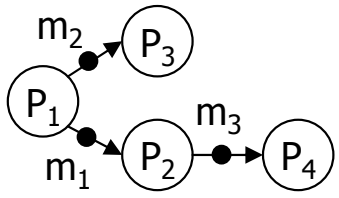
Current solution

Design transformations

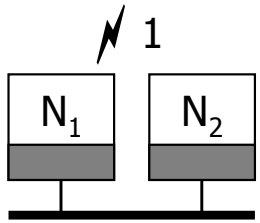


	P_1	P_2	P_3	P_4
Tabu	2	1	0	0
Wait	0	0	2	1

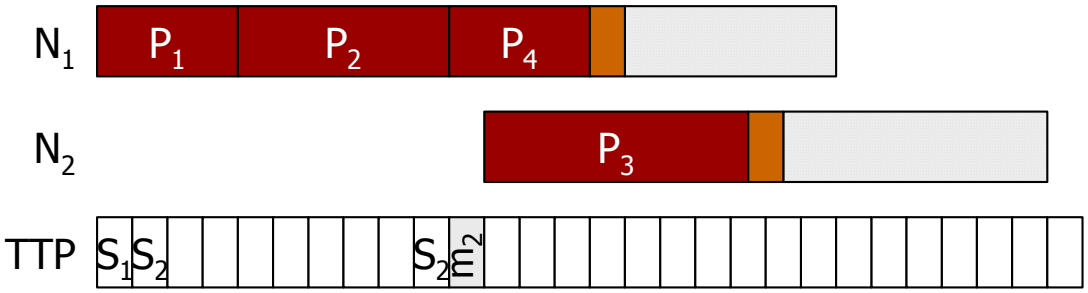
Tabu move & **better** than best-so-far



	N_1	N_2
P_1	40	50
P_2	60	75
P_3	60	75
P_4	40	50



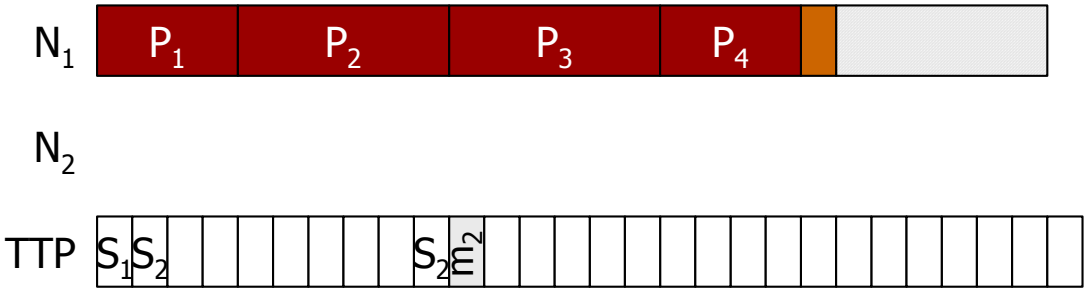
MRX Tabu-Search Example



	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

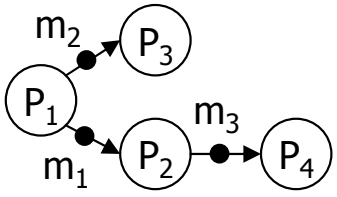
Current solution

Design transformations

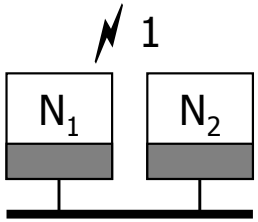


	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

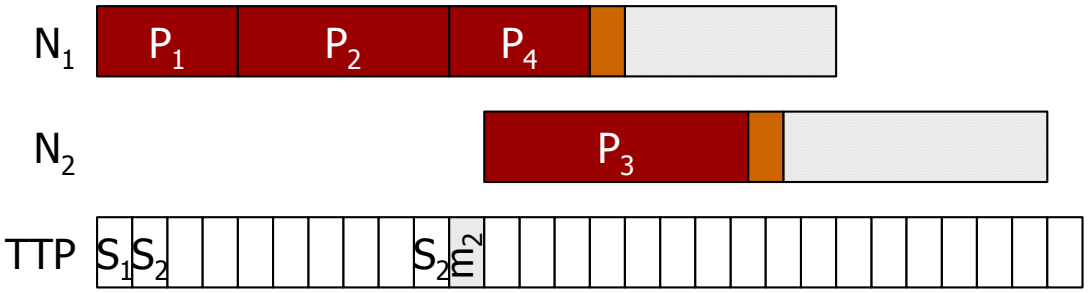
Non-tabu & **worse** than best-so-far



	N_1	N_2
P_1	40	50
P_2	60	75
P_3	60	75
P_4	40	50



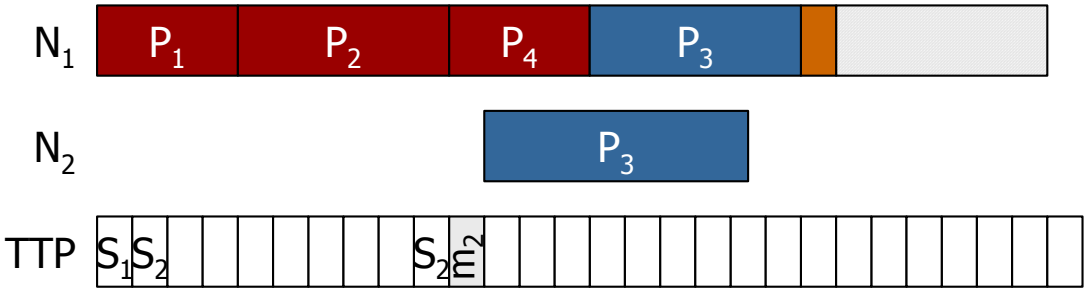
MRX Tabu-Search Example



	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

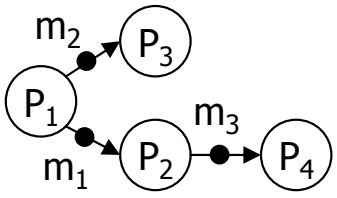
Current solution

Design transformations

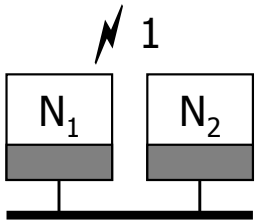


	P_1	P_2	P_3	P_4
Tabu	1	2	0	0
Wait	1	0	1	1

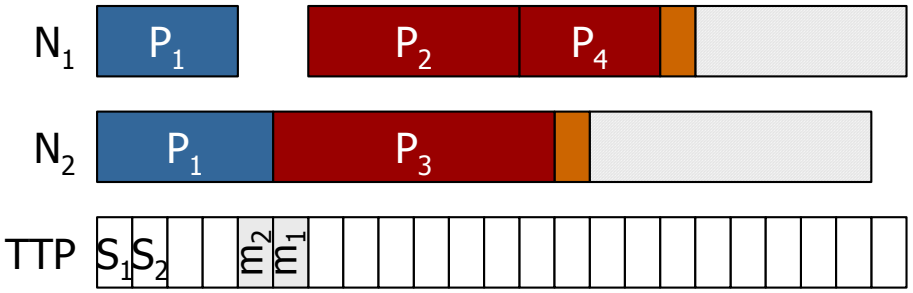
Non-tabu & **worse** than best-so-far



	N_1	N_2
P_1	40	50
P_2	60	75
P_3	60	75
P_4	40	50



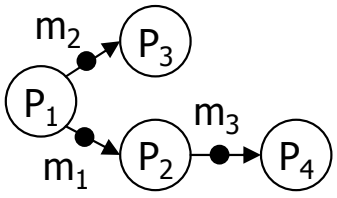
MRX Tabu-Search Example



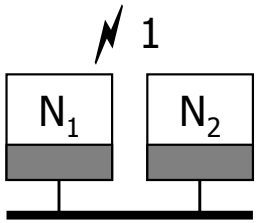
	P_1	P_2	P_3	P_4
Tabu	2	1	0	0
Wait	0	0	2	1

Current solution

Design transformations

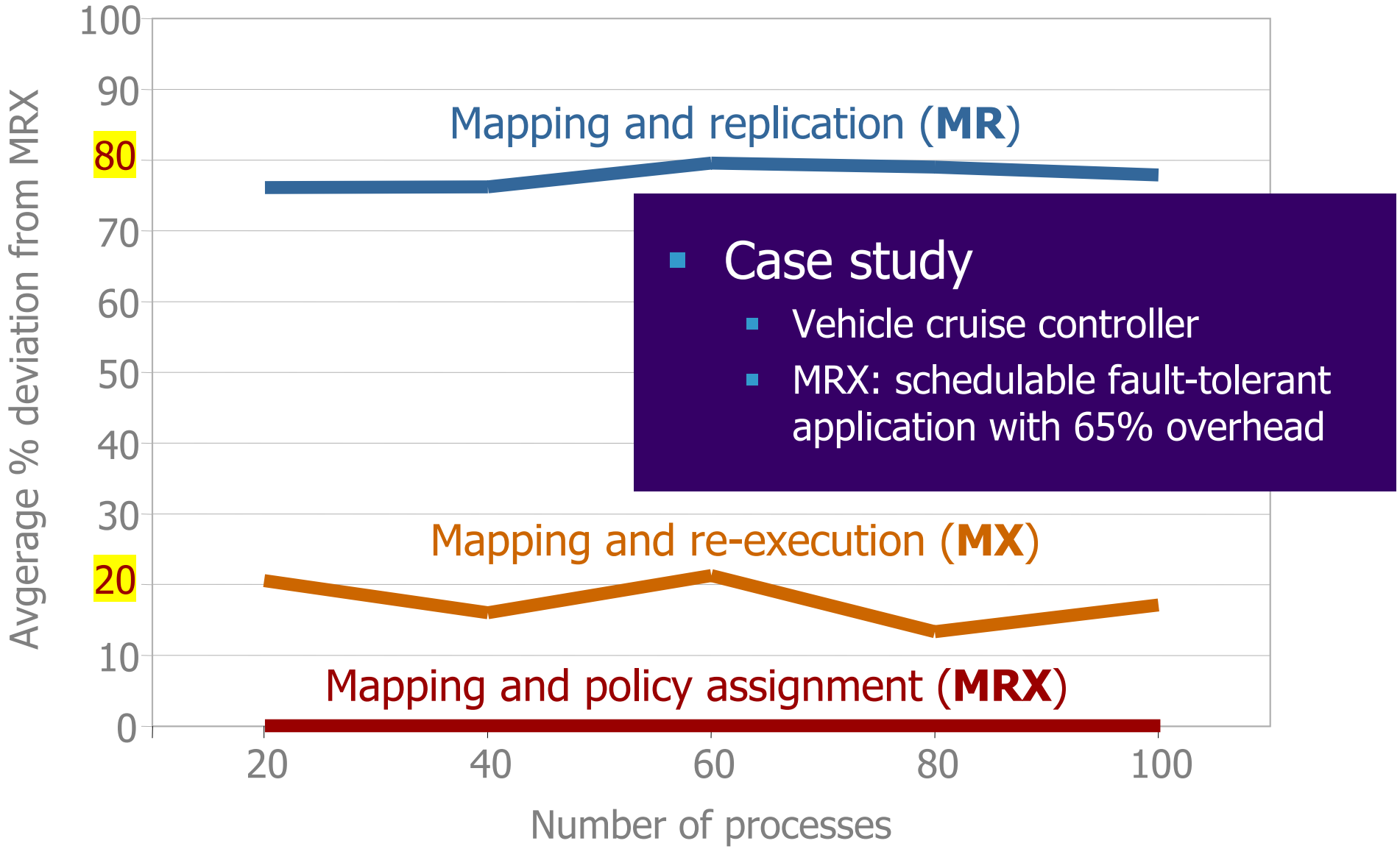


	N_1	N_2
P_1	40	50
P_2	60	75
P_3	60	75
P_4	40	50



Experimental Results

Schedulability improvement under resource constraints



- Case study
  - Vehicle cruise controller
  - MRX: schedulable fault-tolerant application with 65% overhead

Contributions and Message

- Contributions
  - Combined re-execution and replication
  - Optimization algorithms for fault-tolerance policy assignment
  - Efficient contingency schedule generation

Optimization of fault-tolerance policy assignment needed for cost-effective fault tolerance